

REMARKS

Initially, Applicant expresses appreciation to the Examiner for the courtesies extended in the recent in-person interview conducted in connection with this application. The amendments and remarks presented herein are consistent with those discussed in the interview. Accordingly, entry of this amendment and reconsideration of the pending claims is respectfully requested.

The Office Action, mailed October 31, 2007, considered and rejected claims 1-17. Claims 1-9, 11-15 and 17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Grassens* (U.S. Patent No. 7,231,633) in view of *Haikin* (U.S. Patent No. 6,757,893). Claims 10 and 16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Grassens* in view of *Haikin* and *Schmidt* (U.S. Patent No. 4,558,413).¹ Additionally, claims 1-10 and 17 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.² Claim 2 was rejected under 35 U.S.C. § 112, second paragraph, and claims 10 and 17 were objected to for minor informalities which have been corrected in this amendment.

As reflected in the claims listed above, and as discussed above, Applicant's claims relate to storage media and systems for associating original source code with binary code to allow the binary code to be debugged. As reflected in claim 1, for example, a source code file that includes source code and an associated version are stored on a server in a source code file. The source code file is compiled into a binary file and, while doing so, information identifying the location of the source code file and the version of the source code file is extracted. The extracted information is stored in a debug file associated with the binary file. After compiling, an instruction can then be received to debug the binary file. Following such, the extracted information is used from the debug file and the source code file is located and associated with the binary file. Thereafter, the debugging is performed on the binary file with full source code support by correlating lines of the source code with binary instructions in the binary file, and such that the source code file includes only the source code originally used to compute the binary file. Claim 11 recites a system generally capable of performing a method similar to that

¹ Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

² Claims 1-10 have each been amended to recite a computer readable storage media, thereby excluding communication media. Inasmuch as claim 17 has also been cancelled, Applicant respectfully submits that the rejection under 35 U.S.C. § 101 is overcome.

performed upon executing the instructions contained in the computer-readable storage media of claim 1.

As discussed during the interview, while *Grassens* generally relates to a development environment which uses a debugging tool, it fails to disclose or suggest, whether alone or in combination with *Haikin*, each and every element of the claimed invention. For example, among other things, *Haikin* fails to disclose that information identifying a location of a source code file is extracted while compiling the source code, or that debugging is performed after compiling of the source code with full source code support by correlating lines of the source code with binary instructions in a binary file, and particularly such that the source code file includes only the source code originally used to compile the binary file, as recited in combination with the other claim elements.

Specifically, *Grassens* describes a system in which during compiling of the program, debug statements are automatically generated and placed into the program. (Col. 1, ll. 30-35). More particularly, computer source code is compiled and executed and an automated code trace annotator generates annotated source code. (Col. 2, ll. 55-64). The automated code trace annotator does so by systematically stepping through the source code and adding output statements that create an execution log when the annotated code is executed. (Col. 3, ll. 4-10). In this manner, the source code can be annotated and executed to determine a portion of the source code to change. (Col. 3, ll. 11-18). Those changes may then be incorporated into the source code and the code trace annotator can re-execute and may continually debug the source code without having to manually add and remove debugging statements. (Col. 3, ll. 15-23).

Accordingly, *Grassens* describes a system in which source code is compiled and executed, and in which that code is debugged and changed. Notably, however, Applicant's claims recite wherein the source code is compiled into a binary file and the binary file is debugged. In other words, *Grassens* debugs the source code automatically at compile time, where the pending claims recite debugging the binary file after compiling of the source code. Furthermore, while *Grassens* describes keeping track of lines of source code, it does not disclose or suggest that such lines are associated during debugging with specific binary instructions in the binary file. Indeed, *Grassens* has no binary file for association during debugging since the debugging and code modification procedure takes place at the same time.

Applicant respectfully submits that *Haikin* fails to remedy the deficiencies of *Grassens*. In particular, *Haikin* generally relates to a software source code version control system during development and maintenance of software by multiple software developers, and allows version tracking on a line-by-line basis. Thus, *Haikin* describes that versions are attached to source code, but does not disclose that a binary file is associated with the original source code, or that a location of the source code is extracted during compiling of the source code file. Furthermore, while *Haikin* tracks versions on a line-by-line basis, it fails to disclose correlating lines of source code with binary instructions in the binary file as recited in the pending claims.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 533-9800.

Dated this 31st day of March, 2008.

Respectfully submitted,



RICK D. NYDEGGER

Registration No. 28,651

COLBY C. NUTTALL

Registration No. 58,146

Attorneys for Applicant

Customer No. 047973